BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
1	BRS	L1	0	instruction WITH speculative WITH identifier WITH priority WITH request	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:21	
2	BRS	L2	0	instruction WITH speculative WITH identifier WITH priority SAME request	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:21	
3	BRS	L3	0	instruction WITH speculative WITH identifier SAME priority SAME request	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:21	
4	BRS	L4	0	instruction WITH speculative SAME identifier SAME priority SAME request	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:22	
5	BRS	L5	0	(instruction ADJ set) WITH speculative SAME identifier SAME priority SAME request	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:23	
6	BRS	L6	82	(instruction ADJ set) WITH speculative	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:23	
7	BRS	L7	0	6 and (identifier SAME priority SAME request)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:23	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
8	BRS	L8	13	6 and priority and request	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/06/21 11:26	

Terms used
request and **speculative** and **target** and **interconnect** and **memory controller** and **priority**

Found **13,961** of **138,517**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Piranha: a scalable architecture based on single-chip multiprocessing

Luiz André Barroso, Kourosh Gharachorloo, Robert McNamara, Andreas Nowatzyk, Shaz Qadeer, Barton Sano, Scott Smith, Robert Stets, Ben Verghese

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Full text available: pdf(191.10 KB) Additional Information: full citation, abstract, references, citings, index terms

The microprocessor industry is currently struggling with higher development costs and longer design times that arise from exceedingly complex processors that are pushing the limits of instruction-level parallelism. Meanwhile, such designs are especially ill suited for important commercial applications, such as on-line transaction processing (OLTP), which suffer from large memory stall times and exhibit little instruction-level parallelism. Given that commercial applications constitute by fa ...

2 Smart Memories: a modular reconfigurable architecture

Ken Mai, Tim Paaske, Nuwan Jayasena, Ron Ho, William J. Dally, Mark Horowitz

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Full text available: pdf(80.16 KB) Additional Information: full citation, abstract, references, citings, index terms

Trends in VLSI technology scaling demand that future computing devices be narrowly focused to achieve high performance and high efficiency, yet also target the high volumes and low costs of widely applicable general purpose designs. To address these conflicting requirements, we propose a modular reconfigurable architecture called Smart Memories, targeted at computing needs in the 0.1&mgr; technology generation. A Smart Memories chip is made up of many processing tiles, each containing local ...

3 Timestamp snooping: an approach for extending SMPs

Milo M. K. Martin, Daniel J. Sorin, Anatassia Ailamaki, Alaa R. Alameldeen, Ross M. Dickson, Carl J. Mauer, Kevin E. Moore, Manoj Plakal, Mark D. Hill, David A. Wood

November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 28 , 34 Issue 5 , 5

Full text available: pdf(164.27 KB) Additional Information: full citation, abstract, references, citings, index terms

Symmetric muultiprocessor (SMP) servers provide superior performance for the commercial workloads that dominate the Internet. Our simulation results show that over one-third of cache misses by these applications result in cache-to-cache transfers, where the data is

found in another processor's cache rather than in memory. SMPs are optimized for this case by using snooping protocols that broadcast address transactions to all processors. Conversely, directory-based shared-memory systems must indir ...

4 Timestamp snooping: an approach for extending SMPs

Milo M. K. Martin, Daniel J. Sorin, Anastassia Ailamaki, Alaa R. Alameldeen, Ross M. Dickson, Carl J. Mauer, Kevin E. Moore, Manoj Plakal, Mark D. Hill, David H. Wood

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11

Full text available: pdf(1.30 MB) Additional Information: full citation, abstract, references, citings, index terms

Symmetric multiprocessor (SMP) servers provide superior performance for the commercial workloads that dominate the Internet. Our simulation results show that over one-third of cache misses by these applications result in cache-to-cache transfers, where the data is found in another processor's cache rather than in memory. SMPs are optimized for this case by using snooping protocols that broadcast address transactions to all processors. Conversely, directory-based shared-memory systems must indire ...

5 A comparative study of arbitration algorithms for the Alpha 21364 pipelined router

Shubhendu S. Mukherjee, Federico Silla, Peter Bannon, Joel Emer, Steve Lang, David Webb

October 2002 **Proceedings of the 10th international conference on architectural support for programming languages and operating systems**, Volume 30 , 36 , 37 Issue 5 , 5 , 10

Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references

Interconnection networks usually consist of a fabric of interconnected routers, which receive packets arriving at their input ports and forward them to appropriate output ports. Unfortunately, network packets moving through these routers are often delayed due to conflicting demand for resources, such as output ports or buffer space. Hence, routers typically employ *arbiters* that resolve conflicting resource demands to maximize the number of matches between packets waiting at input ports an ...

6 The Stanford FLASH multiprocessor

J. Kuskin, D. Ofelt, M. Heinrich, J. Heinlein, R. Simoni, K. Gharachorloo, J. Chapin, D. Nakahira, J. Baxter, M. Horowitz, A. Gupta, M. Rosenblum, J. Hennessy

April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture**, Volume 22 Issue 2

Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, citings, index terms

The FLASH multiprocessor efficiently integrates support for cache-coherent shared memory and high-performance message passing, while minimizing both hardware and software overhead. Each node in FLASH contains a microprocessor, a portion of the machine's global memory, a port to the interconnection network, an I/O interface, and a custom node controller called MAGIC. The MAGIC chip handles all communication both within the node and among nodes, using hardwired data paths for efficient data moveme ...

7 Using dataflow analysis techniques to reduce ownership overhead in cache coherence protocols

Jonas Skeppstedt, Per Stenström

November 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 6

Full text available: pdf(284.68 KB) Additional Information: full citation, abstract, references, index terms, review

In this article, we explore the potential of classical dataflow analysis techniques in removing overhead in write-invalidate cache coherence protocols for shared-memory multiprocessors. We construct the compiler algorithms with varying degree of sophistication that detect loads followed by stores to the same address. Such loads are marked and constitute a hint to the cache to obtain an exclusive copy of the block so that the subsequent store does not introduce access penalties. The simplest ...

Keywords: cache coherence, dataflow analysis, performance evaluation

8 The Stanford FLASH multiprocessor

Jeffrey Kuskin, David Ofelt, Mark Heinrich, John Heinlein, Richard Simoni, K. Gharachorloo, J. Chapin, D. Nakahira, J. Baxter, M. Horowitz, A. Gupta, M. Rosenblum, J. Hennessy

August 1998 **25 years of the international symposia on Computer architecture (selected papers)**

Full text available: pdf(1.48 MB) Additional Information: full citation, references, index terms

9 Compiler transformations for high-performance computing

David F. Bacon, Susan L. Graham, Oliver J. Sharp

December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms, review

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

10 Parallel architectures: Inferential queueing and speculative push for reducing critical communication latencies

Ravi Rajwar, Alain Kägi, James R. Goodman

June 2003 **Proceedings of the 17th annual international conference on Supercomputing**

Full text available: pdf(568.93 KB) Additional Information: full citation, abstract, references, index terms

Communication latencies within critical sections constitute a major bottleneck in some classes of emerging parallel workloads. In this paper, we argue for the use of Inferentially Queued Locks (IQLs) [31], not just for efficient synchronization but also for reducing communication latencies, and we propose a novel mechanism, Speculative Push (SP), aimed at reducing these communication latencies. With IQLs, the processor infers the existence, and limits, of a critical section from the use of synch ...

Keywords: data forwarding, inferential queueing, synchronization

11 Using "test model-checking" to verify the Runway-PA8000 memory model

Rajnish Ghughal, Abdel Mokkedem, Ratan Nalumasu, Ganesh Gopalakrishnan

June 1998 **Proceedings of the tenth annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(1.14 MB) Additional Information: full citation, references, citings, index terms

12 A survey of commercial parallel processors

Edward Gehringer, Janne Abullarade, Michael H. Gulyn

September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4

Full text available: pdf(2.96 MB) Additional Information: full citation, abstract, citings, index terms

This paper compares eight commercial parallel processors along several dimensions. The processors include four shared-bus multiprocessors (the Encore Multimax, the Sequent Balance system, the Alliant FX series, and the ELXSI System 6400) and four network multiprocessors (the BBN Butterfly, the NCUBE, the Intel iPSC/2, and the FPS T Series). The paper contrasts the computers from the standpoint of interconnection structures, memory configurations, and interprocessor communication. Also, the share ...

13 Algorithms II: Quantifying instruction criticality for shared memory multiprocessors

Tong Li, Alvin R. Lebeck, Daniel J. Sorin

June 2003 **Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(200.96 KB) Additional Information: full citation, abstract, references, index terms

Recent research on processor microarchitecture suggests using instruction criticality as a metric to guide hardware control policies. Fields et al. [3, 4] have proposed a directed acyclic graph (DAG) model for characterizing program microexecutions on uniprocessors. Under such a model, critical path analysis can be applied and instructions' slack values can be used to quantify instruction criticality. In this paper, we extend the uniprocessor DAG model to characterize parallel program executions ...

Keywords: critical path analysis, shared memory multiprocessors, slack

14 Token coherence: decoupling performance and correctness

Milo M. K. Martin, Mark D. Hill, David A. Wood

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(269.08 KB) Additional Information: full citation, abstract, references

Many future shared-memory multiprocessor servers will both target commercial workloads and use highly-integrated "glueless" designs. Implementing low-latency cache coherence in these systems is difficult, because traditional approaches either add indirection for common cache-to-cache misses (directory protocols) or require a totally-ordered interconnect (traditional snooping protocols). Unfortunately, totally-ordered interconnects are difficult to implement in glueless designs. An ideal coherenc ...

15 Memory sharing predictor: the key to a speculative coherent DSM

An-Chow Lai, Babak Falsafi

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture**, Volume 27 Issue 2

Full text available: pdf(122.70 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Recent research advocates using general message predictors to learn and predict the coherence activity in distributed shared memory (DSM). By accurately predicting a message and timely invoking the necessary coherence actions, a DSM can hide much of the remote access latency. This paper proposes the *Memory Sharing Predictors (MSPs),* pattern-based predictors that significantly improve prediction accuracy and implementation cost over general message predictors. An MSP is based on the key ob ...

16 Transactional lock-free execution of lock-based programs

Ravi Rajwar, James R. Goodman

October 2002 **Proceedings of the 10th international conference on architectural support for programming languages and operating systems**, Volume 36 , 30 , 37 Issue 5 , 5 , 10

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings

This paper is motivated by the difficulty in writing correct high-performance programs. Writing shared-memory multi-threaded programs imposes a complex trade-off between programming ease and performance, largely due to subtleties in coordinating access to shared data. To ensure correctness programmers often rely on conservative locking at the

expense of performance. The resulting serialization of threads is a performance bottleneck. Locks also interact poorly with thread scheduling and faults, r ...

17 Architecture and design of AlphaServer GS320

Kourosh Gharachorloo, Madhu Sharma, Simon Steely, Stephen Van Doren

November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 28 , 34 Issue 5 , 5

Full text available: pdf(413.91 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the architecture and implementation of the AlphaServer GS320, a cache-coherent non-uniform memory access multiprocessor developed at Compaq. The AlphaServer GS320 architecture is specifically targeted at medium-scale multiprocessing with 32 to 64 processors. Each node in the design consists of four Alpha 21264 processors, up to 32GB of coherent memory, and an aggressive IO subsystem. The current implementation supports up to 8 such nodes for a total of 32 processors. While s ...

18 Architecture and design of AlphaServer GS320

Kourosh Gharachorloo, Madhu Sharma, Simon Steely, Stephen Van Doren

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11

Full text available: pdf(1.67 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the architecture and implementation of the AlphaServer GS320, a cache-coherent non-uniform memory access multiprocessor developed at Compaq. The AlphaServer GS320 architecture is specifically targeted at medium-scale multiprocessing with 32 to 64 processors. Each node in the design consists of four Alpha 21264 processors, up to 32GB of coherent memory, and an aggressive IO subsystem. The current implementation supports up to 8 such nodes for a total of 32 processors. While s ...

19 SafetyNet: improving the availability of shared memory multiprocessors with global checkpoint/recovery

Daniel J. Sorin, Milo M. K. Martin, Mark D. Hill, David A. Wood

May 2002 **ACM SIGARCH Computer Architecture News**, Volume 30 Issue 2

Full text available: pdf(1.28 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

We develop an availability solution, called *SafetyNet,* that uses a unified, lightweight checkpoint/recovery mechanism to support multiple long-latency fault detection schemes. At an abstract level, *SafetyNet* logically maintains multiple, globally consistent checkpoints of the state of a shared memory multiprocessor (i.e., processors, memory, and coherence permissions), and it recovers to a pre-fault checkpoint of the system and re-executes if a fault is detected. *SafetyNet* e ...

Keywords: availability, shared memory, multiprocessor

20 Using destination-set prediction to improve the latency/bandwidth tradeoff in shared-memory multiprocessors

Milo M. K. Martin, Pacia J. Harper, Daniel J. Sorin, Mark D. Hill, David A. Wood

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(220.76 KB) Additional Information: full citation, abstract, references

Destination-set prediction can improve the latency/bandwidth tradeoff in shared-memory multiprocessors. The destination set is the collection of processors that receive a particular coherence request. Snooping protocols send requests to the maximal destination set (i.e., all processors), reducing latency for cache-to-cache misses at the expense of increased traffic. Directory protocols send requests to the minimal destination set, reducing bandwidth

at the expense of an indirection through the d ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next